

Indexation et recherche par le contenu dans une base d'images fixes : l'intérêt des règles d'association

Anicet Kouomou Choupo, Laure Berti-Équille, Annie Morin

IRISA, Campus universitaire de Beaulieu
F-35042 Rennes Cedex
{akouomou, berti, Annie.Morin}@irisa.fr

Résumé. Les images fixes peuvent, entre autre, être décrites au niveau pixel par des descripteurs visuels globaux de couleur, de texture ou de forme. La recherche par le contenu exploite et combine alors ces descripteurs dont le coût de calcul est d'autant plus important que la taille de la base d'images est grande. Or, un sous-ensemble de descripteurs pourrait suffire à répondre à une recherche par similarité beaucoup plus rapidement, tout en gardant une qualité acceptable des résultats de recherche. Pour cela, nous proposons une méthode de sélection automatique des descripteurs visuels qui exploite les règles d'association pour élaborer des plans d'exécution réduisant le temps de la recherche par le contenu dans de grandes bases d'images. Dans cet article, nous présentons également comment une recherche par le contenu peut être adaptée pour proposer des résultats intermédiaires qui sont fusionnés de façon progressive avec l'avantage pour l'utilisateur, d'une part, de ne pas attendre que toute la base ait été parcourue avant de fournir un résultat et, d'autre part, de lui permettre de stopper la requête en cours d'exécution. Nous évaluons notre méthode comparativement au temps et au résultat d'une recherche séquentielle sur tous les descripteurs de la base.

1. Introduction

La recherche d'informations par le contenu dans de grandes bases de documents multimédias (vidéos, textes, images, ...) exploite l'indexation de descriptions intermédiaires des documents dont les coûts de calculs peuvent être très importants. Pour un même type de données, l'ensemble des descripteurs potentiellement intéressants est souvent très grand. Considérons, pour illustrer ces propos, une base d'images fixes pouvant être décrite de différentes façons, notamment par des descripteurs visuels globaux de couleur, de texture, ou de forme (au niveau pixel). De nombreux descripteurs sont proposés dans la littérature [Manjunath et al., 2002 ; Obeid et al., 2001 ; Tao et Grosky, 1999] dont certains sont standardisés comme MPEG-7. Chacun d'eux est défini selon l'information que l'on souhaite extraire de l'image. On privilégiera par exemple un descripteur de forme si l'on souhaite retrouver toutes les images contenant un clavier d'ordinateur. Par contre, un descripteur de couleur sera indiqué si l'on recherche des images de coucher du soleil, par exemple. Et que dire d'une image contenant à la fois un coucher de soleil et un clavier d'ordinateur ? Il est possible, dans ce cas, d'utiliser à la fois un descripteur de forme et un de couleur. Mais, selon les objectifs de recherche de l'utilisateur, seul le descripteur de forme pourrait suffire à

Règles d'association et recherche par le contenu

discriminer et filtrer les images les plus similaires. Dans ce cas, il est inutile de vouloir faire la recherche sur tous les descripteurs pour toutes les données.

Un des inconvénients majeurs de la recherche par le contenu est qu'elle est généralement menée de façon exhaustive sur la totalité de la base, sans doute du fait des schémas d'indexation relativement peu efficaces sur de grands volumes de données. Ceci se traduit par des temps d'attente inacceptables pour l'utilisateur, jusqu'à ce que toutes les images de la base soient comparées avec l'image-requête selon une mesure de similarité. De plus, tout arrêt brutal de la requête en cours d'exécution a pour conséquence la perte de toutes les informations, obligeant l'utilisateur à relancer sa recherche. Par opposition aux bases de données traditionnelles [Manolescu, 2002] [Gounaris et al., 2002], très peu de travaux ont été menés sur l'optimisation et le traitement adaptatif de la recherche par le contenu dans un contexte multimédia. À notre connaissance, seuls quelques travaux sur les requêtes progressives ont été très récemment proposés [Kiranyaz et Gabbouj, 2004], mais ils se limitent à lancer périodiquement une même sous-requête sur des portions d'une base de documents multimédias (images ou vidéos) sans ciblage *a priori*, ni fusion entre des descripteurs hétérogènes.

Dans cet article, nous proposons d'abord une stratégie de sélection automatique des critères de recherche par le contenu, fondée sur l'utilisation des règles d'association afin de réduire le nombre de descripteurs pour une recherche donnée. Nous présentons ensuite comment la recherche par le contenu peut être adaptée pour proposer des résultats intermédiaires qui sont progressivement fusionnés avec l'avantage, pour l'utilisateur, d'une part, de ne pas attendre que toute la base ait été parcourue et, d'autre part, de stopper la progression de la requête en cours d'exécution. Nous utilisons une base d'images fixes pour évaluer notre méthode en comparant le temps et les résultats obtenus par rapport à ceux d'une recherche séquentielle sur tous les descripteurs de la base.

Notre travail combine deux techniques de fouille : le clustering et la recherche de règles d'association qui permettent d'améliorer notablement le temps de la recherche dans de grandes base d'images sans véritablement dégrader les résultats que l'on obtiendrait par une recherche séquentielle.

Cet article est structuré de la manière suivante : dans un premier temps nous faisons, en section 2, une synthèse des travaux liés à l'extraction des règles d'association et à leur utilisation pour l'indexation et pour la recherche par le contenu dans des bases d'images. Nous présentons ensuite, en section 3, le principe de description des images fixes, ainsi que notre méthode de sélection automatique des critères de recherche (*i.e.*, des descripteurs visuels à utiliser), puis nous formalisons, en section 4, l'adaptation progressive de la recherche par le contenu intégrant notamment l'usage dynamique des règles d'association. En section 5, nous évaluons expérimentalement notre approche. La section 6 conclut l'article et décrit nos perspectives de travail.

2. État de l'art

Après une brève présentation des techniques d'indexation utilisées pour organiser les bases d'images, nous nous intéressons en particulier aux travaux qui ont consisté à utiliser ou à adapter les règles d'association à la recherche par le contenu.

Les techniques d'indexation adaptées à l'organisation d'une base d'images fixes peuvent être classées en deux grands types :

– les techniques basées sur le partitionnement de données. Elles sont toutes dérivées du *R-Tree* [Guttman, 1984] et procèdent par un regroupement des vecteurs (un descripteur étant un vecteur de réels pouvant atteindre plusieurs centaines de dimensions) selon leur proximité relative dans l'espace. Les vecteurs sont englobés dans une forme géométrique simple (hyper-rectangle, par exemple) et le tout est organisé sous forme d'un arbre dans lequel les vecteurs sont stockés dans les feuilles alors que les formes englobantes sont stockées dans les nœuds.

– les techniques basées sur le partitionnement de l'espace. Elles ne tiennent pas compte de la distribution des vecteurs et font un partitionnement de l'espace en cellules plus ou moins régulières. Les cellules générées peuvent être organisées soit de façon arborescente comme dans le cas du *k-d-Tree* [Bentley, 1979], soit par une table de hachage comme c'est le cas du *GridFile* [Nievergelt et al., 1984].

Un panorama détaillé de toutes ces techniques d'indexation est présenté dans [Berrani et al., 2002].

Alors que les techniques d'indexation restent adaptées à des vecteurs de petite dimension, les performances d'une recherche par le contenu se faisant par calcul d'une distance aux plus proches voisins sont très rapidement dégradées lorsque la dimension des vecteurs des descripteurs devient grande. On est donc amenés à définir de nouvelles structures d'indexation adaptées aux vecteurs de grande dimension ainsi que des méthodes de recherche approximative pour accélérer la recherche. Certaines méthodes de recherche approximative élaborent des modèles probabilistes pour contrôler la précision des résultats [Berrani et al., 2003]. D'autres travaux exploitent les règles d'association dans un contexte de recherche par le contenu multimédia. Parmi ceux-ci, citons les travaux de Djeraba [Djeraba, 2003], Bouet et Khenchaf [Bouet et Khenchaf, 2002] et Zaïane [Zaïane et al., 2000] qui nous permettent de resituer notre approche.

Djeraba utilise une technique d'indexation voisine de la nôtre en ce sens qu'elle combine la construction des clusters et la détermination des règles d'association [Djeraba, 2003]. Sa base de travail est formée de plusieurs sous-groupes d'images classées et étiquetées thématiquement (animaux, plantes, ...) et l'idée consiste à caractériser ces groupes (ou clusters d'images) par des règles d'association définissant une sémantique entre les valeurs des descripteurs et les labels des classes d'images. Lors de la recherche, une analyse de la requête à partir des règles d'association oriente le choix du sous-groupe d'images. Le travail de Djeraba a pour objectif l'amélioration de la qualité de la recherche. Il ne s'intéresse pas au problème de fusion des résultats lorsque la recherche se fait suivant plusieurs descripteurs visuels. Le temps de la recherche n'est pas évalué. De plus, l'approche est contrainte par la construction semi-automatique d'un dictionnaire nécessitant une labellisation sémantique des clusters. Notre approche ne limite pas le nombre ainsi que le type de descripteurs utilisables et aucun étiquetage n'est nécessaire puisque nous travaillons au niveau pixel, c'est-à-dire « sans sémantique ajoutée ». Notre stratégie combine uniquement le calcul des clusters et l'extraction des règles d'association. Elle est donc entièrement non supervisée et convient aux grandes bases d'images.

Le travail de Bouet et Khenchaf [Bouet et Khenchaf, 2002] tente de capturer le niveau sémantique par la découverte des relations existantes entre les images d'une base de données. Ces images sont regroupées en clusters pour chacun des descripteurs disponibles. Les règles d'association, utilisées à titre descriptif, sont calculées pour faciliter l'exploitation des

Règles d'association et recherche par le contenu

clusters. Leur antécédent correspond à une valeur de propriété visuelle et leur conséquent à un cluster. La principale différence entre l'approche de Bouet et Khenchaf et la nôtre se situe au niveau du rôle que jouent les règles d'association dans le processus de recherche. Dans [Bouet et Khenchaf, 2002], les règles sont proposées pour cibler les clusters d'intérêt au moyen des mesures de qualité qui les accompagnent. Les règles sont dans ce cas utilisées pour sélectionner les clusters. Dans notre approche, la sémantique des règles diffère et nous utilisons les règles sur les clusters déjà sélectionnés pour réduire les critères de recherche.

Les nombreux algorithmes d'extraction des règles d'association proposés dans la littérature, tels que *Apriori* [Agrawal et al., 1993], *Pascal* [Bastide et al., 2002], sont très efficaces pour des bases de données transactionnelles dont la structure est bien définie. Mais, leur application aux images nécessite des adaptations. Une idée consiste à transformer les images et à appliquer les techniques classiques d'extraction de règles. Une image sera par exemple identifiée à une transaction et les différentes régions la constituant à des objets [Ordonez et Omiecinski, 1999].

La répétition d'objets peut sous certaines conditions être porteuse d'informations. Un grand nombre de régions de couleur bleue dans une image donne par exemple une idée de l'intensité de la couleur bleue. La définition classique des règles d'association ne tient pas compte de cette possibilité de répétition et il est nécessaire dans ce cas de développer de nouveaux formalismes. C'est ainsi que Zaïane et al. [Zaïane et al., 2000] proposent une définition de *règle d'association avec objets récurrents* comme une implication de la forme :

$$\alpha_1 P_1 \wedge \alpha_2 P_2 \wedge \dots \wedge \alpha_n P_n \rightarrow \beta_1 Q_1 \wedge \beta_2 Q_2 \wedge \dots \wedge \beta_m Q_m$$

où $P_i, i \in [1..n]$ et $Q_j, j \in [1..m]$ sont des descripteurs d'images, et α_i, β_j sont des entiers indiquant le nombre d'occurrences des descripteurs. L'algorithme *MaxOccur* décrit dans [Zaïane et al., 2000] est une adaptation de *Apriori* pour l'extraction des règles d'association avec objets récurrents.

Notre approche utilise des règles d'association pour l'indexation et pour la recherche par le contenu dans une base d'images fixes. Nous utilisons plusieurs descripteurs et les images sont regroupées en clusters pour chaque descripteur. Nous identifions ensuite une image à une transaction et un numéro de cluster pour un descripteur donné à un attribut. Le problème de répétition d'objets ou d'attributs ne se pose pas puisque les clusters sont identifiés de manière unique pour chaque descripteur.

3. Sélection automatique des critères de recherche

Nous présentons maintenant le principe de description des images fixes. Puis nous décrivons notre méthode de sélection automatique des critères de recherche par le contenu. Ce système, introduit pour la première fois dans [Kouomou-Choupo et al., 2004], se décompose en :

- une phase d'indexation, et repose sur la génération de relations sous forme de règles d'association entre les groupes de descripteurs
- une phase de recherche, et repose sur l'utilisation des règles pour choisir parmi les critères disponibles ceux qui sont jugés les plus pertinents pour la recherche envisagée (en terme de qualité du résultat et de performance de la requête).

3.1 Description d'images fixes

La description d'une image a pour but de rassembler tous les éléments nécessaires à la caractérisation de l'image dans un contexte d'utilisation donné. La nature et le type de descripteurs proposés par la communauté du traitement d'images dépendent complètement et précisément du but de la recherche (par exemple, un descripteur particulier pourra être conçu pour retrouver le visage de G. Bush). Quatre niveaux de description ont cependant été proposés [Zhang et al., 2001] : au niveau du pixel, de l'objet, au niveau sémantique, et au niveau connaissance.

Notre étude est volontairement limitée à la description visuelle des images qui sont traitées comme des ensembles de pixels. Dans un cadre général, étant donnée une image I , une description de I consiste à trouver un vecteur $d=f(I)$ qui résume les caractéristiques de l'image I . Le descripteur d de l'image I est un vecteur de réels ou d'entiers de dimension n et f la fonction de calcul du descripteur. Les images au niveau du pixel peuvent être décrites par des descripteurs globaux, par des descripteurs locaux ou encore des descripteurs *ad hoc* (par exemple pour la détection de visages). Les descripteurs locaux sont une façon assez fine et précise de décrire les images, accroissant ainsi le pouvoir de reconnaissance lors de la recherche par le contenu tout en restant robuste aux occultations partielles, aux variations des conditions de production de l'image et à l'orientation de celle-ci [Smeulders et al., 2000]. En revanche, ils sont coûteux en temps de calcul et demandent beaucoup d'espace de stockage [Amsaleg et Gros, 2001]. Nous avons ici limité notre étude à l'usage des descripteurs globaux et, dans notre cadre expérimental, nous en avons choisi 5 pour décrire respectivement : la couleur (*ColorLayout*, *ScalableColor*), la texture (*HomogeneousTexture*, *EdgeHistogram*) et la forme (*RegionShape*). Ces descripteurs sont proposés dans le standard MPEG-7 [Manjunath et al., 2002].

3.2 Processus d'indexation

Le processus d'indexation se fait hors-ligne. Son point de départ est une base d'images fixes (de type tout venant) à partir de laquelle on calcule l'ensemble des valeurs des descripteurs retenus pour sa caractérisation. La base d'images est ensuite organisée en clusters pour chaque type de descripteurs calculés.

Soit $D = \{d_1, d_2, \dots, d_m\}$ l'ensemble de m descripteurs pouvant être calculés par le système. Un cluster est identifié par un numéro et le descripteur pour lequel il est calculé. Nous désignons par $(n_j(I), d_j)$ le cluster d'appartenance de l'image I pour le descripteur d_j et nous associons à I une transaction dont les attributs sont les clusters auxquels l'image appartient.

Une base d'images notée B peut donc être décrite par $B = \{I | I = \{(n_1(I), d_1), (n_2(I), d_2), \dots, (n_m(I), d_m)\}\}$. Il est possible d'appliquer un algorithme de calcul de motifs fréquents (tel que *Apriori*) sur la base B ainsi décrite et d'extraire des règles d'association pour obtenir des implications (avec un ou plusieurs éléments en partie gauche) de la forme :

$$r : (n_i, d_i) [\wedge (n_2, d_2) \wedge \dots \wedge (n_p, d_p)] \rightarrow (n_k, d_k) < supp, conf > \quad [1]$$

avec $d_i \neq d_j$ si $i \neq j$; $i, j \in \{1, 2, \dots, p\} \cup \{k\}$

où *supp*, *conf* sont le support et la confiance de la règle r , exprimés en pourcentages. Les seuils minimaux de support et de confiance sont fixés et permettent de juger la qualité des règles découvertes.

Règles d'association et recherche par le contenu

Nous avons travaillé sur une base de 30411 images décrites par les descripteurs *ColorLayout*, *ScalableColor*, *HomogeneousTexture*, *EdgeHistogram* et *RegionShape* que nous notons respectivement *CLD*, *SCD*, *HTD*, *EHD* et *RSD*. Au terme du processus d'indexation, nous obtenons sur la base *B* un ensemble de clusters pour chaque descripteur d_i et un ensemble de règles obtenu avec *Apriori* [Agrawal et al., 1993] qui explicitent les relations entre les différents clusters obtenus. En voici deux exemples :

R1 : (23, RSD) \wedge (1, HTD) \wedge (23, EHD) \rightarrow (8, CLD) <0.1, 55.2>
R2 : (19, HTD) \wedge (0, SCD) \rightarrow (29, CLD) <0.1, 66.1>

Supposons que la règle R2 soit appliquée. Sa sémantique sera interprétée comme suit : une image qui appartient aux clusters (19, HTD) et (0, SCD) appartient aussi au cluster (29, CLD) dans 66.1 % des cas. Il est donc inutile dans la recherche de parcourir tous les 3 clusters de la règle d'association. On pourrait se limiter aux clusters en partie gauche de la règle puisque leur parcours induit en partie celui du cluster de la partie droite. L'avantage immédiat de cette restriction est le gain en temps de recherche. Nous détaillons dans le paragraphe suivant la sémantique des règles obtenues ainsi que l'usage qui en est fait.

3.3 Processus de recherche par le contenu

Dans le cas d'une recherche par l'exemple (*QBE* - *Query-By-Example*), l'utilisateur soumet au système une image-requête. Son but est de retrouver toutes les images similaires à l'image-requête selon certains critères visuels. L'utilisateur peut savoir choisir des descripteurs visuels tels que ceux proposés dans la norme MPEG-7 comme critères de recherche. Mais, il arrive le plus souvent qu'il n'ait aucune idée des descripteurs les plus discriminants pour sa requête. Comme nous l'avons déjà mentionné, dans bien des cas, il s'avère inutile de faire la recherche sur tous les critères déjà précalculés sur l'ensemble des images. Pour ce type de recherche, notre travail est, en particulier, de proposer les descripteurs qui seraient les plus adaptés et de leur affecter des priorités utiles pour optimiser l'exécution de la requête.

Nous nous plaçons dans un contexte de recherche progressive par le contenu où des résultats intermédiaires sont retournés à l'utilisateur au fur et à mesure de la progression de la recherche. Nous désignons par phase de recherche l'étape de traitement dont la fin correspond à l'envoi d'un résultat intermédiaire à l'utilisateur. La numérotation des phases commence à 1. Étant donnée une image-requête I_q , le système calcule, pour chaque descripteur d_j , une mesure de sélectivité des clusters (définie ci-après), puis détermine, à une phase donnée, les clusters dans lesquels rechercher les plus proches voisins de I_q .

Notons $TopC(M, i)$ l'ensemble des M clusters de meilleure sélectivité sélectionnés à la phase i . Il s'agit ensuite d'exploiter les règles d'association entre les clusters de $TopC(M, i)$ et ceux des phases précédentes de façon à éliminer de $TopC(M, i)$ certains clusters et les descripteurs correspondants pour accélérer les premiers temps de la recherche. Soit R l'ensemble des règles calculées dans le processus d'indexation (section 3.2). Pour toute règle $r \in R$, notons C_r l'ensemble des clusters présents dans l'expression de la règle r (à la fois en partie droite et gauche).

Notre hypothèse initiale est la suivante : pour les règles dont le support et la confiance sont supérieurs à un seuil donné, à une phase k de la recherche, une règle r est sélectionnée si :

$$C_r \subseteq \bigcup_{i=1..k} TopC(M,i) .$$

Intuitivement à l'issue du processus d'indexation et d'extraction des règles définies au paragraphe précédent, une règle de la forme [1] exprime, si elle est sélectionnée, que les plus proches voisins de l'image-requête I_q appartenant à tous les clusters de la partie gauche de la règle [1] sont probablement aussi présents dans la partie droite de la règle. Notre stratégie de sélection de descripteurs consiste donc, en priorité, à ignorer systématiquement pour la recherche toutes les parties droites des règles sélectionnées et donc, tous les clusters de descripteurs correspondants.

4. Adaptation progressive des requêtes pour la recherche par le contenu

La recherche par le contenu nécessite généralement le parcours exhaustif de toute la base d'images avant de pouvoir fournir un résultat à l'utilisateur. Face à ce constat, notre objectif est d'améliorer, au niveau logique, le traitement d'une requête par le contenu en proposant : 1) une stratégie de planification des requêtes qui définisse un ordre optimal de parcours des clusters d'images, 2) la possibilité de fournir à l'utilisateur des résultats intermédiaires qui soient rafraîchis au fur et à mesure du parcours progressif de la base et 3) la possibilité à l'utilisateur de stopper l'exécution de la requête s'il juge que le dernier résultat intermédiaire retourné est pertinent pour sa recherche.

Le principe général, présenté dans la Figure 1, consiste à lancer, à l'instant initial T_o , une image-requête I_q qui est décomposée en une série de sous-requêtes sur des portions de la base d'images (c'est-à-dire des clusters de descripteurs) ciblées selon leur sélectivité, et les priorités éventuellement données par les règles d'association qui ont été préalablement extraites lors de la phase d'indexation.

La sélectivité d'un cluster dépend de la taille du cluster considéré et de sa proximité à l'image-requête décrite selon un descripteur donné (par exemple, dans la Figure 1, le cluster C_{11} sera le plus proche de $D_1(I_q)$, description de l'image-requête selon le descripteur D_1). Les sous-requêtes fournissent des résultats intermédiaires qui sont progressivement mis à jour et raffinés au cours du temps global d'exécution de la requête sur la totalité de la base. L'avantage est ici de fournir, sans attendre, des résultats intermédiaires à l'utilisateur au cours de l'exécution de la requête, qu'il pourra d'ailleurs stopper (T_{stop}).

Après la fusion des résultats de recherche sur chaque cluster, les résultats intermédiaires de chaque sous-requête sont envoyés (à T_1, T_2, T_3). Lorsque toute la base a été parcourue, le dernier résultat intermédiaire est fusionné avec les résultats des derniers clusters parcourus et il est envoyé comme résultat final à l'utilisateur.

Règles d'association et recherche par le contenu

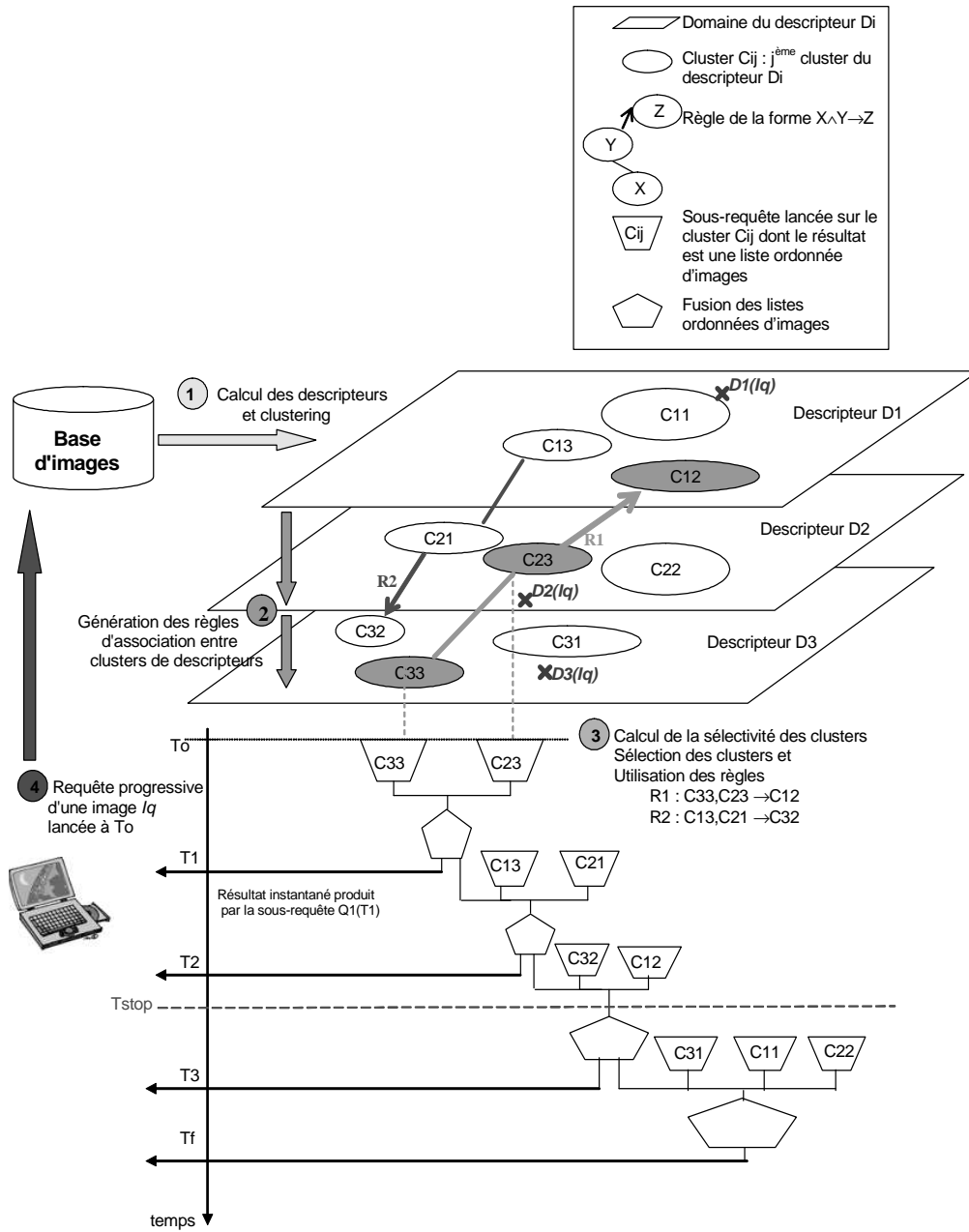


FIG. 1 - Exemple de recherche progressive

4.1 Définitions formelles

Base d'images

Soit D l'ensemble des descripteurs pouvant être calculés, la base d'images B est définie comme l'union des k clusters C_{ij} de chaque descripteur d_i présent de la base ($d_i \in D$). Plus formellement,

$$\forall d_i \in D, i = 1..card(D), B = \bigcup_{j=1..k} C_{ij} \quad [2]$$

Une image appartient à la base B si sa description selon chaque descripteur présent dans la base appartient à un cluster.

$$I \in B \text{ si } \forall d_i \in D, \exists C_{ij}, d_i(I) \in C_{ij}$$

Requête par l'exemple

Une requête par l'exemple $I_q = (n_q, O_q, W_q, D_q)$ dans l'espace multidimensionnel S de dimension d_s est composée des informations suivantes :

- un nombre de points n_q dans l'image-requête I_q
- un ensemble de n_q points $O_q = \{O_q^{(1)}, \dots, O_q^{(n_q)}\}$ dans l'espace multidimensionnel des descripteurs
- un ensemble de n_q poids $W_q = \{w_q^{(1)}, \dots, w_q^{(n_q)}\}$, le $i^{\text{ème}}$ poids $w_q^{(i)}$ étant associé au $i^{\text{ème}}$ objet $O_q^{(i)}$ ($0 \leq w_q^{(j)} \leq 1, \sum_{i=1}^{n_q} w_q^{(i)} = 1$)
- une fonction de distance D_q , qui, pour un point O donné dans l'espace S , calcule la distance entre la requête et le point. On suppose que D_q est une distance pondérée L_p , i.e., pour une valeur donnée de p , la distance entre deux points T_1 et T_2 dans S est définie telle que :¹

$$D_q = \left[\sum_{j=1}^{d_s} \mu_q^{(j)} (|T_1[j] - T_2[j]|)^p \right]^{1/p} \quad [3]$$

où $\mu_q^{(j)}$ est le poids associé à la $j^{\text{ème}}$ dimension de S ($0 \leq \mu_q^{(j)} \leq 1, \sum_{j=1}^{d_s} \mu_q^{(j)} = 1$).

D_q spécifie quelle mesure L_p utiliser (i.e., la valeur de p) et les valeurs des poids par dimension. Nous utilisons la fonction de distance D_q pour construire la fonction de distance agrégée $D_q(I_q, O)$ entre les objets multiples de la requête O_q et l'objet O (dans S).

$D_q(I_q, O)$ est une fonction agrégée des distances entre O et les objets $O_q^{(i)} \in O_q$:

$$D_q(I_q, O) = \sum_{i=1}^{n_q} w_q^{(i)} D_q(O_q^{(i)}, O) \quad [4]$$

Nous utilisons une somme pondérée comme fonction d'agrégation mais tout autre fonction peut être utilisée tant qu'elle est pondérée et monotone.

¹ Cette hypothèse est générale pour la plupart des fonctions de distance utilisées. La distance de Manhattan, la distance Euclidienne, et la distance au rectangle englobant sont des cas particuliers de distance L_p .

Règles d'association et recherche par le contenu

Phase de recherche

La $i^{\text{ème}}$ phase de recherche correspond à un calcul de recherche de similarité sur l'ensemble des clusters de rang i pour chaque descripteur.

Sélectivité des clusters

La sélectivité d'un cluster C pour une requête par l'exemple donnée I_q est une fonction combinant la proximité du cluster à l'image-requête (précédemment définie comme la fonction de distance D_q), la taille du cluster et la densité en son centre.

La mesure de sélectivité d'un cluster C par rapport à une requête par l'exemple I_q , est notée $Selectivity(C, I_q)$ et elle est définie telle que :

$$Selectivity(C, I_q) = \alpha \left(1 - \frac{dist(I_q, C)}{\max_j (dist(I_q, C_j))} \right) + \beta \left(1 - \frac{size(C)}{\max_j (size(C_j))} \right) + \gamma \cdot density(C)$$

avec $\alpha \geq 0, \beta \geq 0, \gamma \geq 0$ et $\alpha + \beta + \gamma = 1$

[5]

$size(C)$ est la taille du cluster C , $dist(I_q, C)$ est la distance de I_q au centre du cluster C et $density(C)$ est la densité du cluster C en son centre.

Dans la suite de ce travail, nous poserons $\gamma = 0$ de façon à limiter ici notre étude aux seuls critères relatifs à la taille du cluster et à la distance de son centre à une image requête. La notion de densité est l'objet d'un travail en cours d'expérimentation.

Les paramètres α, β, γ sont positifs et choisis tels que $0 \leq Selectivity(C, I_q) \leq 1$.

L'ordre de traitement des clusters sera défini pour une requête par l'exemple donnée I_q selon que leur sélectivité est proche de 1.

Règle d'association entre clusters

Une règle d'association entre plusieurs clusters de descripteurs $C_{ij}, C_{i'j'}, C_{i''j''}$, est notée de la façon suivante :

$$r : C_{ij} \wedge \dots \wedge C_{i'j'} \rightarrow C_{i''j''} < \text{supp}, \text{conf} >$$

[6]

avec $i, i', i'' = 1..card(D)$, $i \neq i' \neq i''$ et $j, j', j'' = 1..k$

La sémantique de la règle r est la suivante : une image dont la description appartient aux clusters $C_{ij}, \dots, C_{i'j'}$, appartient probablement au cluster $C_{i''j''}$ selon les taux de support et de confiance $supp$ et $conf$.

Sélection des règles d'association entre clusters

Pour une requête par l'exemple donnée I_q , les clusters sont sélectionnés selon leur mesure de sélectivité pour la $i^{\text{ème}}$ phase de recherche. L'ensemble des $Top M$ clusters à cette phase est alors noté : $TopC_M^i(I_q)$.

L'ensemble des descripteurs correspondant est noté : $TopD_M^i(I_q)$.

Une règle d'association est sélectionnée pour la requête par l'exemple I_q à l'étape k de la recherche si son support et sa confiance sont supérieurs à des seuils fixés et si l'ensemble des clusters composant la règle (en parties gauche et droite), noté C_r , est tel que :

$$C_r \subseteq \bigcup_{i=1..k} Top C_M^i(I_q) \quad [7]$$

Lorsqu'une règle est sélectionnée, le traitement de la requête ignore dans un premier temps les clusters présents en partie droite de la règle.

Plan d'exécution d'une requête par l'exemple

Rappelons que $D = \{d_1, \dots, d_m\}$ est l'ensemble des descripteurs et qu'une base d'images est organisée en clusters suivant chaque descripteur. Un plan d'exécution logique P pour une requête par l'exemple I_q est composé de la liste des clusters C_{ij} à parcourir pour retrouver les images les plus similaires selon des priorités définies pour le traitement des descripteurs D de la base. Il se décompose en plusieurs sous-plans tels que :

$$P(I_q, D) \leftarrow P_1(I_q, d_1) \circ P_2(I_q, d_2) \circ \dots \circ P_i(I_q, d_i) \circ \dots \circ P_m(I_q, d_m) \quad [8]$$

$P_i(I_q, d_i)$ correspond au traitement des clusters du descripteur d_i (c'est-à-dire la recherche des plus proches voisins de I_q dans chaque cluster de d_i).

Il y a $\prod_{i=1..m} (n_c(d_i))!$ plans possible avec $n_c(d_i)$, le nombre de clusters pour le descripteur d_i .

$P(I_q, D)$ peut être réécrit au niveau des ensemble de clusters pour chaque descripteur tel que:

$$P(I_q, D) \leftarrow P_1(I_q, \{C_{11}, \dots, C_{1k}\}) \circ P_2(I_q, \{C_{21}, \dots, C_{2k}\}) \circ \dots \circ P_m(I_q, \{C_{m1}, \dots, C_{mk}\})$$

ou comme un plan de requête minimale pour chaque cluster tel que:

$$P(I_q, D) \leftarrow P_1(I_q, C_{11}) \circ \dots \circ P_k(I_q, C_{1k}) \circ P_{k+1}(I_q, C_{21}) \circ \dots \circ P_{2k}(I_q, C_{2k}) \circ \dots \circ P_{(m-1)k+1}(I_q, C_{m1}) \circ \dots \circ P_{mk}(I_q, C_{mk})$$

avec k clusters par descripteur et m descripteurs.

Requête progressive par l'exemple

Supposons un ordre partiel sur les instants initial, intermédiaires et final de l'exécution de la requête tel que : $T_0 < T_1 < T_2 < \dots < T_f$.

Une requête progressive, notée $\vec{Q}(T_0, freq)$, soumise à l'instant initial T_0 , et terminée à l'instant T_f , est une requête par l'exemple dont les résultats intermédiaires produits par chaque sous-requête (appelés résultats instantanés) sont régulièrement fusionnés et envoyés à l'utilisateur aux instants $T_i \in]T_0, T_f[$ tout au long de l'exécution de la requête selon une fréquence, notée $freq$.

pour $freq = 1$,

$$\vec{Q}(T_0, 1) = Q_1(T_1) \circ Q_2(T_2) \circ \dots \circ Q_{N_c}(T_f) \text{ avec } N_c, \text{ le nombre total de clusters.}$$

Une sous-requête minimale est une recherche des plus proches voisins de l'image-requête au sein d'un cluster de descripteur.

Fréquence d'une requête progressive

La fréquence d'une requête progressive dépend du nombre de résultats intermédiaires requis et du nombre total de clusters à traiter :

Règles d'association et recherche par le contenu

$$freq = \frac{N_i + 1}{N_c} \quad [9]$$

avec N_c : le nombre total de clusters et N_i : le nombre de résultats instantanés requis ($1 \leq N_i \leq N_c - 1$).

Pour une fréquence $freq = 1$, les résultats instantanés sont envoyés à l'utilisateur après fusion des résultats de chaque requête minimale. La requête progressive correspondante est telle que :

$$\vec{Q}(T_0, 1) = Q_1(T_1) \circ Q_2(T_2) \circ \dots \circ Q_n(T_f)$$

Dans la Figure 1, la fréquence est $freq = 4/9$, avec 3 résultats intermédiaires et 9 sous-requêtes minimales (c'est-à-dire 9 clusters). Les premiers résultats intermédiaires sont envoyés à l'utilisateur après fusion des résultats de deux requêtes minimales. Parmi les quatre plans d'exécution possibles qui correspondent à la requête progressive, celui présenté en Figure 1 est le suivant :

$$\vec{Q}(T_0, 4/9) = (((Q_1 \circ Q_2)(T_1) \circ (Q_3 \circ Q_4))(T_2) \circ (Q_5 \circ Q_6))(T_3) \circ (Q_7 \circ Q_8 \circ Q_9)(T_f)$$

Résultat instantané

Un résultat instantané d'une requête progressive \vec{Q} , noté $InstantResult(Q)$, est le résultat d'une sous-requête sur une partie de la base ; il est issu de la fusion des résultats instantanés précédents.

$$(InstantResult(Q_i))^{new} = (InstantResult(Q_{i-1}))^{old} \circ InstantResult(Q_i)$$

Résultat final

Le résultat final d'une requête progressive \vec{Q} est le résultat obtenu :

- soit lorsque l'intégralité de la base d'images a été parcourue pour répondre à la requête initiale (c'est-à-dire quand tous les clusters ont été parcourus)
- soit par l'arrêt volontaire de la requête par l'utilisateur.

4.2 Sélection des plans

Pour une requête donnée, le système va générer tous les plans d'exécution possibles pour retrouver les images les plus similaires à l'image-requête. La figure 2 rappelle les étapes de traitement d'une requête progressive.

Soit Q une requête sur la base d'images et P_Q l'ensemble de tous les plans pour Q . L'espace de recherche pour trouver les N meilleurs plans pour Q est l'ensemble de tous les sous-ensembles de taille N de P_Q .

$$\mathcal{S}(Q) = \{ P' \subseteq P_Q \mid |P'| = N \}$$

Dans notre étude, nous nous sommes volontairement limités au choix d'un des meilleurs plans possibles en sélectionnant à chaque phase de recherche et pour chaque descripteur, le cluster non encore parcouru et de sélectivité la plus élevée (Tableau 1).

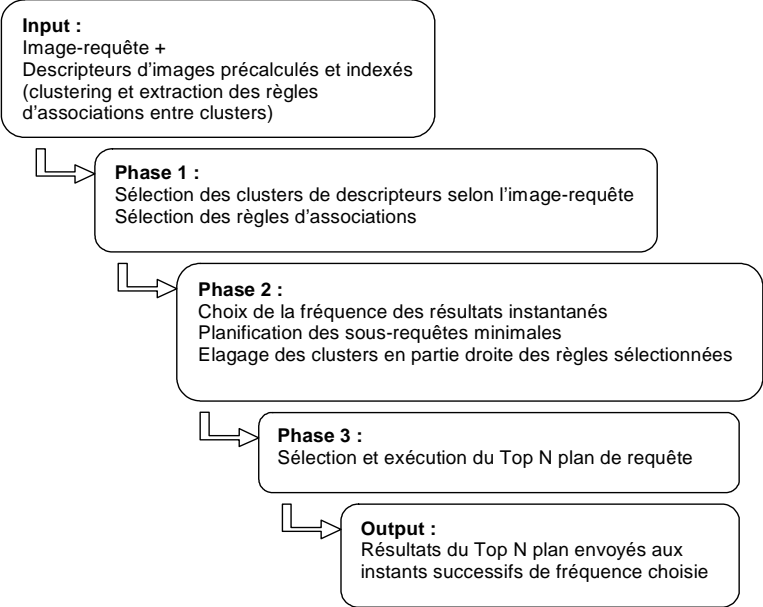
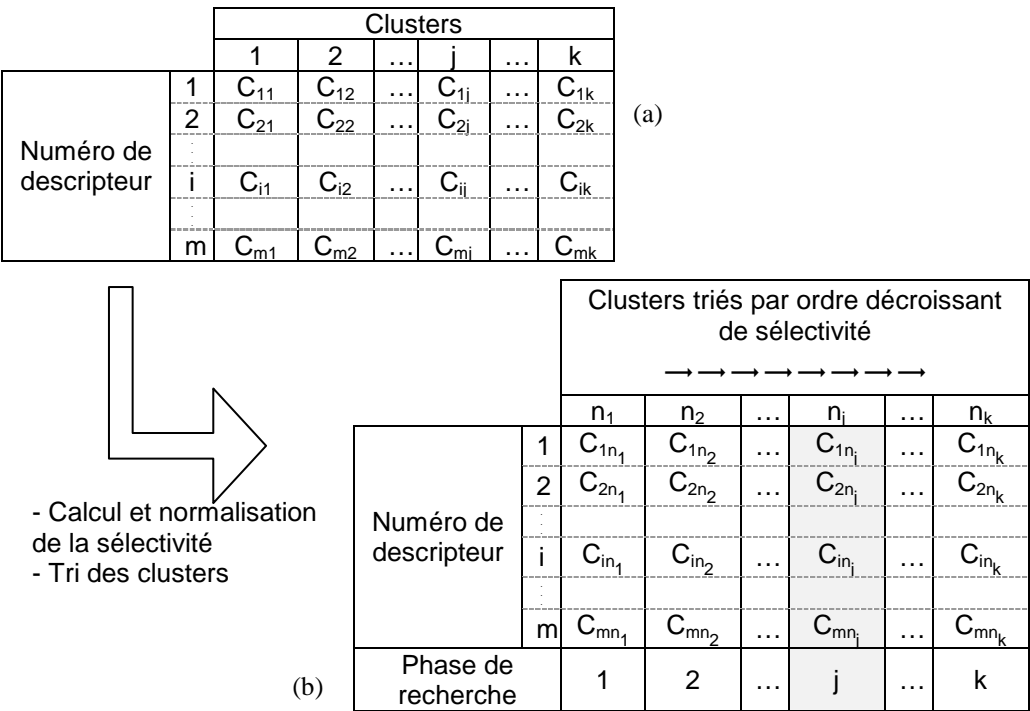


FIG. 2 - Planification d'une requête progressive



TAB 1 – Grille de planification d'une exécution d'un plan de requête progressive

Règles d'association et recherche par le contenu

Le tableau 1 détaille le plan d'exécution retenu par notre système de recherche. Le cluster numéro j du descripteur i est noté C_{ij} avant le tri (Tableau 1(a)). Après le tri, un cluster sera noté C_{in_j} ou n_j désignera le cluster en position j pour le descripteur i . Si I_q désigne une image-requête, les clusters parcourus à la phase de recherche j sont C_{in_j} tels que :

$$Selectivity(C_{in_j}, I_q) = \max_{l \in \{1, \dots, k\} - \{n_1, \dots, n_{j-1}\}} Selectivity(C_{il}, I_q), \quad i = 1..m$$

Ces clusters parcourus à l'étape j figurent dans la partie grisée du tableau 1(b). Ils sont triés suivant chaque descripteur. La normalisation de la sélectivité est en cours d'expérimentation. Elle permet de trier les clusters tous descripteurs confondus. Le parcours des clusters les plus sélectifs entraînant ainsi le choix des descripteurs les plus intéressants.

4.3 Fusion des résultats instantanés

Sans nuire à la généralité, supposons que la recherche se fait, à une phase donnée, sur un sous-ensemble $\{d_1, d_2, \dots, d_{m_1}\}$ de m_1 descripteurs sélectionnés parmi les m disponibles ($m_1 \leq m$). $m_1 = m$ si aucune règle d'association n'est sélectionnée.

Notons l_j les listes ordonnées, résultats de la recherche suivant les descripteurs sélectionnés d_j , avec $1 \leq j \leq m_1$. Il s'agit de les fusionner pour obtenir le résultat final également sous forme d'une liste ordonnée. Nous définissons pour une image I un score de fusion par la formule suivante :

$$S_f(I) = \frac{1}{2m_1} \left(\sum_{j=1}^{m_1} (f_{d_j}(I) + S_{d_j}(I)) \right) \quad [10]$$

où $f_{d_j}(I) = 1$ si $I \in l_j$ et 0 sinon. $S_{d_j}(I)$ est le score de I pour le descripteur d_j . Il est proportionnel à la similarité entre I et l'image requête I_q . La similarité entre deux images se traduit par la distance entre leur vecteur de descripteurs respectifs. Nous utilisons les distances proposées par MPEG-7 dans la définition des descripteurs. Ces distances sont pour la plupart des variantes de la distance euclidienne. Nous supposons que $S_{d_j}(I) \in [0,1]$. En particulier, $S_{d_j}(I) = 0$ si $I \notin l_j$. Dans ces conditions, le score fusionnel $S_f(I)$ est compris entre 0 et 1. Il tient compte de la fréquence d'apparition de l'image I dans les listes l_j et de son score lorsqu'elle apparaît comme résultat dans une liste.

Considérons à titre d'exemple deux listes ordonnées et de même longueur :

$$l_1 = ((I_1, 0.90), (I_3, 0.70), (I_5, 0.50))$$

$$l_2 = ((I_3, 0.80), (I_4, 0.40), (I_6, 0.30)).$$

Chaque composante de liste est un couple (image, score). Nous nous proposons de calculer la liste finale l_f obtenue par fusion de l_1 et l_2 .

D'après la formule [10], on a pour l'image I_1 :

$$S_f(I_1) = \frac{1}{2 \times 2} ((1 + 0.90) + (0 + 0)) \approx 0.48.$$

Après les calculs sur toutes les images, on obtient la liste finale suivante :

$$l_f = ((I_3, 0.88), (I_1, 0.48), (I_5, 0.38), (I_4, 0.35), (I_6, 0.33)).$$

La fusion des listes de résultats issues de la comparaison de descripteurs hétérogènes est un problème complexe [Fagin et al., 2003] [Berretti et al., 2003] et nous envisageons d'autres fonctions de fusion.

5. Expérimentations

La méthode de sélection automatique de critères de recherche présentée précédemment est implémentée en C++ sous Linux. Nous travaillons pour l'instant avec une base de 30411 images fixes, mais nous envisageons prochainement le traitement de plusieurs centaines de milliers d'images. Nous exploitons les descripteurs MPEG-7 de couleur (*Color Layout*, *ScalableColor*), de texture (*HomogeneousTexture*, *EdgeHistogram*) et de forme (*RegionShape*). Pour chacun des 5 descripteurs MPEG-7 utilisés, nous regroupons les images en clusters avec un algorithme de type *k-means*. Le problème du calcul automatique du nombre de clusters reste entier. Dans ce travail, nous partons du constat que les clusters très gros (en terme de nombre d'images qu'ils contiennent) apparaissent le plus souvent en partie droite de règle de façon à garder élevée la valeur de la confiance. Ainsi, nous adoptons une approche expérimentale dans laquelle l'homogénéité de la taille des clusters détermine le choix de leur nombre. La détermination du nombre de clusters reste cependant difficile. Une tentative de résolution du problème est présentée dans [Fernandez et al., 2002]. Mais la complexité du regroupement en clusters est plus grande.

Dans ce paragraphe, nous nous intéressons dans un premier temps à l'organisation de la base d'images fixes par des clusters et par des règles d'association. Nous étudions ensuite l'usage des règles d'association dans un contexte de recherche progressive par le contenu. Les images de notre base sont fournies par une agence photographique. Nous n'avons pas de détails sur leur provenance pour envisager un plan robuste d'échantillonnage. Les images sont « du tout venant » et l'idée consiste à bien les organiser pour en faciliter une recherche par l'exemple. Nous choisissons pour cette raison 100 images requêtes aléatoirement, toutes déjà présentes dans la base d'images et nous recherchons pour chacune d'elles les 15 plus proches voisins, puis nous mesurons la qualité et le temps moyens de la recherche. Tous les résultats présentés dans cette étude sont comparés à ceux obtenus par une recherche séquentielle. À l'heure actuelle, il est très difficile d'évaluer, « dans l'absolu », la qualité des résultats d'une recherche par l'exemple sur des images, car elle dépend en grande partie d'objectifs implicites de l'utilisateur dont l'image-requête n'est pas toujours porteuse de façon auto-suffisante. C'est pourquoi, d'abord motivés par les problèmes de performance, nous définissons la notion de qualité de la recherche, par rapport à ce que l'on obtient par une recherche séquentielle, en comptant le nombre d'images communes entre chaque résultat instantané obtenu par notre approche et le résultat final obtenu au terme de la recherche séquentielle sur toute la base d'images et suivant tous les descripteurs. Dans cette mesure de qualité, nous ne nous intéressons pas à l'ordre des images dans la liste mais tout simplement à leur présence qui serait déjà satisfaisante pour un utilisateur final.

5.1 Expérience 1 : choix des paramètres d'indexation

Le but est d'étudier les paramètres d'indexation tels que le nombre de clusters, les seuils du support et de confiance des règles d'association. L'utilisation de *k-means* comme algorithme de calcul des clusters implique effectivement le choix préalable du nombre de

Règles d'association et recherche par le contenu

clusters. Ce choix est difficile, mais peut être guidé par les règles d'association dont la découverte dépend de la taille des clusters. Nous calculons, pour une valeur fixée du nombre de clusters, l'écart type de la taille de tous les clusters exprimé en nombre d'images. Cette mesure donne une idée de la taille des clusters par rapport au nombre moyen d'images par cluster. La figure 3(a) présente les variations de l'écart type par rapport au nombre de clusters pour chacun des 5 descripteurs utilisés. Elle montre que pour des valeurs inférieures à 10 du nombre de clusters, on obtient des clusters très disproportionnés en taille. Le nombre de cluster doit être suffisamment grand pour que la taille ne s'écarte pas beaucoup de la moyenne. Ce constat est vrai pour tous les descripteurs qui se comportent de la même manière dans la répartition des images de chaque cluster. Le nombre de clusters pour tous les descripteurs, qu'il soit le même ou pas, ne modifie pas notre principe d'indexation et de recherche d'images fixes. Nous pouvons donc poser comme hypothèse de travail, sans nuire à la généralité, que le nombre de clusters est le même pour tous les descripteurs. Dans la suite de nos expérimentations, nous le fixons à 30, car d'après l'expérience reportée dans la figure 3(a), 30 est la valeur où l'écart type de la taille des clusters se stabilise pour l'ensemble des descripteurs.

Les règles d'association sont calculées sur les clusters de descripteurs avec l'algorithme *Apriori* [Agrawal et al., 1993]. Le seuil de confiance est fixé à 50% pour garantir la robustesse des règles. Quant au seuil du support, nous le choisissons expérimentalement. La figure 3(b) donne les variations du nombre de règles calculées en fonction du seuil du support. On obtient en général des règles de support très faible. Un nombre élevé de clusters entraîne la réduction du support des règles. Si l'on suppose par exemple que tous les 30 clusters contiennent exactement le même nombre d'images, c'est-à-dire $\frac{30411}{30}$, alors le support

d'une règle sera au plus de $\frac{\left(\frac{30411}{30}\right) \times 100}{30411} \approx 3,33\%$ sous l'hypothèse que les parties gauche et

droite des règles contiennent les mêmes images. Les règles donnent néanmoins une idée du nombre d'images qui se retrouvent dans un même cluster pour plusieurs descripteurs. Dans l'objectif de générer un grand nombre de règles, nous avons fixé le seuil du support à 0,033. Une règle sera donc retenue si sa confiance est d'au moins 50 % et si les parties gauche et droite des règles ont au moins 10 images communes. Dans ces conditions, 30 clusters étiquetés de 0 à 29 sont calculés pour tous les descripteurs et 279 règles sont produites au terme de la phase d'indexation.

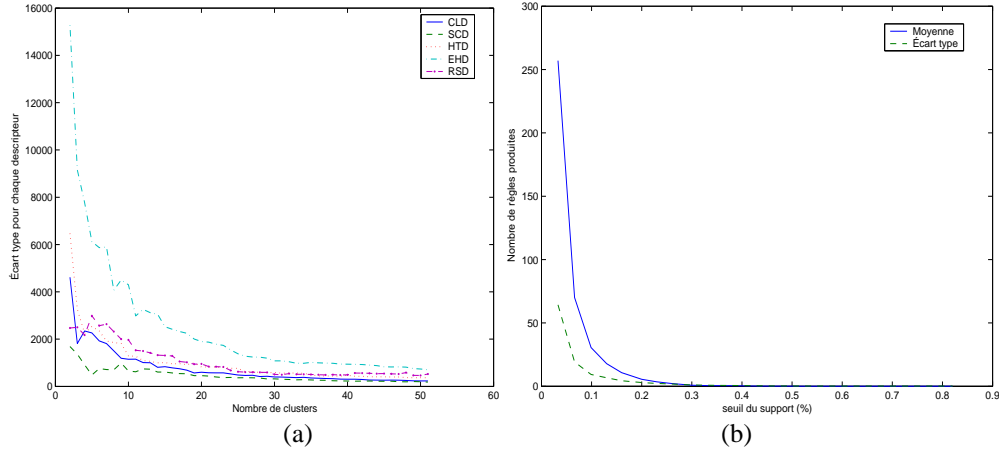


FIG. 3 - Paramètres d'indexation : nombre de clusters (a) support des règles (b)

5.2 Expérience 2 : usage des règles d'association

Le calcul de similarité est fait sur les clusters de descripteurs ordonnés selon la proximité de leur centre à l'image-requête I_q . À la première phase de recherche, le traitement de la requête I_q est mené sur les clusters de premier rang pour chaque descripteur (i.e. $TopC_M^1(I_q)$). À la phase de recherche suivante, le traitement de la requête est fait sur les clusters de second rang et le résultat est fusionné au résultat intermédiaire obtenu à l'étape précédente et mis à jour. La base est organisée en 30 clusters pour chaque descripteur, et la requête aura donc 30 phases de recherche. Pour une requête progressive, les résultats intermédiaires peuvent être envoyés à la fin de chaque phase de recherche. Dans cette expérience, nous nous proposons d'explorer l'usage dynamique des règles d'association : les règles sont sélectionnées tout au long de l'exécution de la requête. Cette approche requiert le parcours de tout l'ensemble des règles d'association chaque fois que la requête est soumise au système. À la 12^{ème} phase de recherche, la qualité du résultat intermédiaire obtenu par l'usage dynamique des règles d'association est presque la même que celle du résultat final obtenu à la fin d'une recherche séquentielle. La figure 4 montre que la perte relative de qualité sur le résultat final obtenu par l'usage dynamique des règles d'association est de $\frac{1 \times 100}{15} \approx 6,67\%$

et le gain relatif de performance (en temps d'exécution de la requête) est de $\frac{(4,6 - 3,55) \times 100}{4,6} \approx 22,9\%$.

D'après cette expérimentation, l'organisation d'une base d'images fixes en clusters suivant chaque descripteur offre un cadre de recherche progressive dans laquelle le résultat final peut être obtenu sans qu'il soit nécessaire de parcourir tous les clusters disponibles. Cette expérimentation montre en outre que l'usage des règles d'association altère la qualité des résultats partiels, mais cette altération est compensée par le gain en temps correspondant.

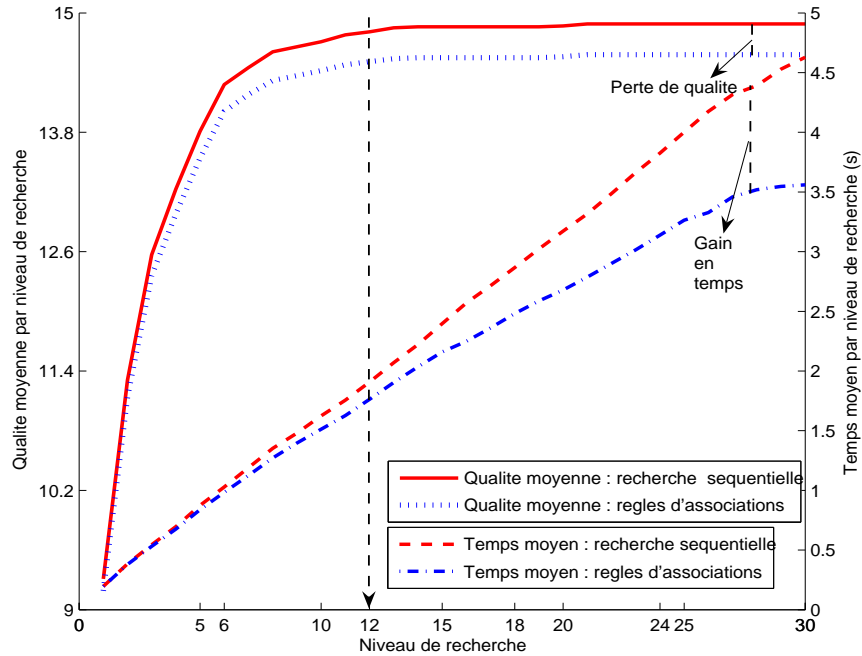


FIG. 4 - Usage dynamique des règles d'association pour le traitement des requêtes progressives

5.3 Expérience 3 : sélection des clusters

Nous étudions dans cette expérience la mesure de sélectivité afin de montrer l'importance de l'ordre des clusters dans l'exécution d'une requête progressive. À partir de la définition de la mesure de sélectivité donnée en section 4.1, nous limitons l'étude à $\gamma=0$. Ainsi, la mesure de sélectivité combine la distance du centre du cluster à l'image-requête et la taille du cluster. Cette mesure privilégie les clusters les plus petits et les plus proches de la requête.

Pour $\alpha=1$, $\beta=0$ et $\gamma=0$, les clusters sont ordonnés selon la proximité de leur centroïde à l'image-requête, les résultats obtenus sont montrés dans l'expérience précédente. Les courbes de qualité et de temps d'exécution pour les différentes valeurs de α , β ($\gamma=0$), sont données dans la Figure 5. Les valeurs $\alpha=0$, $\beta=1$, $\gamma=0$ impliquent la sélection des clusters exclusivement basée sur la taille des clusters. Dans ce cas, le temps d'exécution augmente faiblement, mais la convergence des résultats est ralentie par rapport au cas de sélection sur le critère de proximité ($\alpha=1$, $\beta=0$, $\gamma=0$). Nous observons (Figure 5) qu'à temps égal, la vitesse de convergence des résultats partiels est meilleure pour ($\alpha=1/4$, $\beta=3/4$, $\gamma=0$). On peut naturellement en conclure que la sélection des clusters sur le critère de taille réduit le temps partiel de recherche mais le critère de proximité contribue à préserver une forte convergence des résultats intermédiaires (*i.e.*, une relativement bonne qualité des résultats partiels comparés au résultat final de la recherche séquentielle).

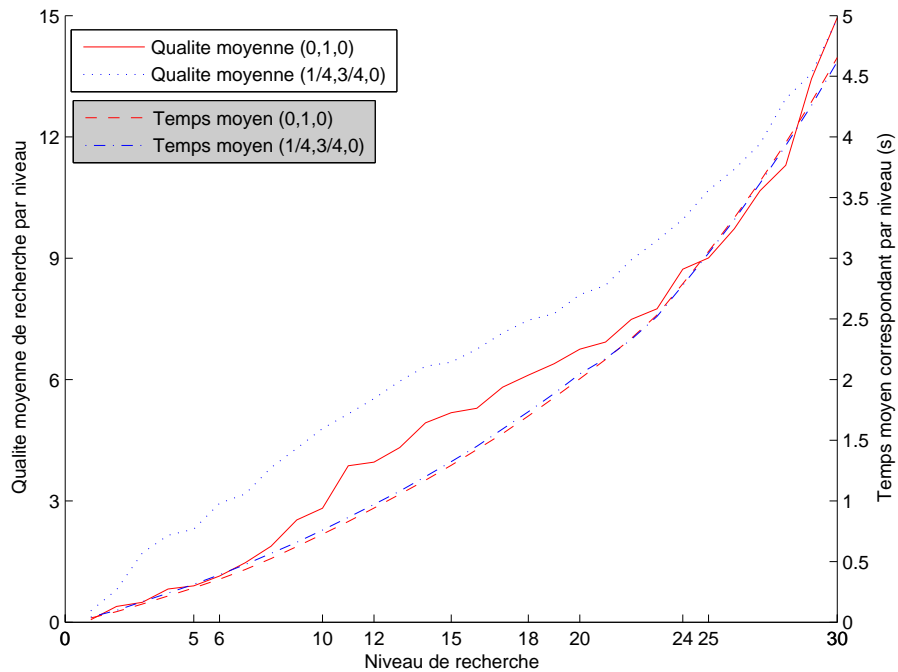


FIG. 5 - Mesure de sélectivité pour le traitement des requêtes progressives

6. Conclusion et perspectives

Nous présentons une stratégie de sélection automatique des critères de recherche par le contenu sur une base d'images. Cette stratégie est fondée sur le clustering des valeurs de descripteurs des images et sur l'utilisation des règles d'association entre ces clusters permettant de définir des priorités pour le traitement des clusters et améliorer le temps d'exécution d'une recherche de similarité entre l'image-requête et la base d'images. Nous introduisons une mesure de sélectivité des clusters de descripteurs, puis nous en étudions les propriétés dans le but de réduire les délais d'attente de l'utilisateur tout en préservant la qualité relative des résultats de la recherche comparée à une approche séquentielle. Tous nos résultats sont comparés à la qualité et au temps d'exécution d'une recherche séquentielle sur une base de 30411 images fixes et 5 descripteurs MPEG-7. Les résultats montrent que d'une part, la recherche progressive simple permet d'avoir presque la totalité du résultat final en la moitié du temps global de recherche et d'autre part, les règles d'association accélèrent davantage le processus de recherche. L'usage des règles d'association est donc intéressant comme moyen de sélection et de réduction des critères de recherche.

L'idée de ce travail est de pouvoir à terme organiser et interroger de très grandes bases d'images fixes en adaptant la sélection des critères de recherche. Dans ce cadre, nous envisageons des expérimentations sur une base plus conséquente (ayant plusieurs centaines de milliers d'images) et de plus nombreux descripteurs. Nous serons ainsi amenés à raffiner

Règles d'association et recherche par le contenu

notre mesure de sélectivité des clusters en précisant notamment l'évaluation de la densité d'un cluster en son centre et périphérie. Nous serons éventuellement contraints de changer la méthode actuelle de détermination de clusters (*k-means*) qui impose le choix préalable du nombre de clusters et risque de s'avérer lente pour le processus d'indexation. De plus, l'ajout d'une image entraîne la description de celle-ci ainsi que le parcours de la base entière pour la mise à jour de l'ensemble des règles d'association. Il est donc judicieux de s'orienter vers des techniques de clustering et vers des mesures de sélection de règles qui améliorent le caractère incrémental du processus.

Références

- [Agrawal et al., 1993] R. Agrawal, T. Imielinski, et A. Swami. Mining Association Rules Between Sets of Items in Large Databases. *ACM SIGMOD Int. Conf. on Management of Data*, 1993, p. 207-216.
- [Amsaleg et Gros, 2001] L. Amsaleg et P. Gros. Content-Based Retrieval Using Local Descriptors: Problems and Issues from a Database Perspective. *Pattern Analysis and Applications, Special Issue on Image Indexation*, vol. 4, 2001, p. 108-124.
- [Bastide et al., 2002] Y. Bastide, R. Taouil, N. Pasquier, G. Stumme, et L. Lakhal. Pascal : un algorithme d'extraction des motifs fréquents. *Technique et science informatique*, vol. 21, n°1, 2002, p. 65-95.
- [Bentley, 1979] J.-L. Bentley. Multidimensional Binary Search in Database Applications. *IEEE Transactions on Software Engineering*, vol. 4, n°5, 1979, p. 333-340.
- [Berrani, et al., 2002] S.-A. Berrani, L. Amsaleg, et P. Gros. Recherche par similarité dans les bases de données multidimensionnelles : panorama des techniques d'indexation. *RSTI - Ingénierie des systèmes d'information. Bases de données et multimédia*, vol. 7, n°5-6, 2002, p. 9-44.
- [Berrani et al., 2003] S.-A. Berrani, L. Amsaleg, et P. Gros. Approximate Searches: k-Neighbors + Precision. *Proc. of the 12th Int. Conf. on Information and Knowledge Management*, 2003, p. 24-31.
- [Berretti et al., 2003] S. Berretti, A. Del Bimbo, et P. Pala. Merging results for distributed content based image retrieval. *Proc. of the Int. workshop on Multimedia Information Systems (MIS'03)*, Ischia, 2003.
- [Bouet et Khenchaf, 2002] M. Bouet et A. Khenchaf. Traitement de l'information multimédia: Recherche du média image. *RSTI -Ingénierie des systèmes d'information. Bases de données et multimédia*, vol. 7, n° 5-6, 2002, p. 65-90.
- [Djeraba, 2003] C. Djeraba. Association and Content-Based Retrieval. *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, n° 1, 2003, p. 118-135.
- [Fagin et al., 2003] R. Fagin, R. Kumar, et D. Sivakumar. Efficient similarity search and classification via rank aggregation. *Proc. of the Int. ACM SIGMOD Conf. on Management of Data*, San Diego, California, USA, pp 301-312, 2003.
- [Fernandez et al., 2002] G. Fernandez, A. Meckaouche, P. Peter, et C. Djeraba. Intelligent Image Clustering. *Lecture Notes in Computer Science*, Vol. 2490, 2002, p. 406-419.
- [Gounaris et al., 2002] A. Gounaris, N. Paton, A. Fernandes, et R. Sakellariou. Adaptive query processing. *BNCOD 2002, LNCS 2405*, pp.11-25, 2002.
- [Guttman, 1984] A. Guttman. R-trees: A dynamic Index Structure for Spatial Searching, *Proc. of the ACM SIGMOD Int. Conf. on Management of Data*, 1984, p. 47-57.

- [Kiranyaz et Gabbouj, 2004] S. Kiranyaz et M. Gabbouj. A novel multimedia retrieval technique: progressive query (why wait ?), *Proc. of the 5th International Workshop on Image Analysis for Multimedia Interactive Services*, Portugal, April 2004.
- [Kouomou-Choupo et al., 2004] A. Kouomou-Choupo, A. Morin, et L. Berti-Équille. Recherche dans de grandes bases d'images fixes : une nouvelle approche guidée par les règles d'association, *Revue RNTI-E-2 (Actes EGC'2004)*, CÉPADUÉS, 2004, p. 65-70.
- [Manjunath et al., 2002] B.-S. Manjunath, P. Salembier, et T. Sikora. Introduction to MPEG-7, John Wiley & Sons, 2002.
- [Manolescu, 2002] I. Manolescu. Adaptive and self-tuning query processing. *EDBT Summer School*, 2002.
- [Nievergelt et al., 1984] J. Nievergelt, H. Hinterberger, et K.-C. Sevcik. The GridFile: An Adaptable, Symmetric Multikey File Structure. *ACM Transactions on Database Systems*, vol. 9, n° 1, 1984, p. 38-71.
- [Obeid et al., 2001] M. Obeid, B. Jedynak, et M. Daoudi. Image Indexing and Retrieval using Intermediate Features, *Proc. of the 9th ACM/ICM*, 2001, p. 531-533.
- [Ordonez et Omiecinski, 1999] C. Ordonez et E. Omiecinski. Discovering Association Rules based on Image Content. *Proc. of the IEEE Advances in Digital Libraries Conf. (ADL'99)*, 1999, p. 38-49.
- [Smeulders et al., 2000] A.-W.-M. Smeulders, M. Worring, S. Santini, A. Gupta, et R. Jain. Content-Based Image Retrieval at the End of the Early Years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, n° 12, 2000, p. 1349-1380.
- [Tao et Grosky, 1999] Y. Tao et W. Grosky. Object-Based image retrieval using point feature maps. *Proc. of the Int. Conf. on Database Semantics (DS-8)*, 1999, p. 59-73.
- [Zaïane et al., 2000] O. Zaïane, J. Han, et H. Zhu. Mining Recurrent Items in Multimedia with Progressive Resolution Refinement. *Proc. of the 16th IEEE Int. Conf. on Data Engineering (ICDE'00)*, 2000, p. 461-476.
- [Zhang et al., 2001] J. Zhang, W. Hsu, et M.-L. Lee. Image Mining: Issues, Frameworks and Techniques. *Proc. of the 2nd Int. Workshop on Multimedia Data Mining*, 2001, p. 13-20.

Summary

Still images can be described at the pixel level by global visual features of color, texture or shape. Content-based retrieval then uses and combines these features whose computing cost is becoming as more important as the database size grows. But a subset of features could be enough to answer a query-by-example much more quickly while keeping an acceptable quality of query results. We then propose a way of selecting visual features using association rules. The method is designed to allow query execution plans that speed up content-based retrieval in very large still image databases. In this paper, we present a strategy for adapting the content-based retrieval and proposing instantaneous results which are progressively merged with the advantage for the user, on one hand, not to wait the end of the processing of the entire base and on the other hand to be able to modify or stop the execution of the current query. We evaluate our method by a comparison with a sequential search on all the features of the database.

RNTI - E -